

SELRand

IEC 61131 Library for ACSELERATOR RTAC® Projects

SEL Automation Controllers

Table of Contents

Section 1: SELRand	
Introduction.....	3
Supported Firmware Versions	3
Functions	3
Benchmarks.....	5
Examples	6
Release Notes.....	9

RTAC LIBRARY

SELRand

Introduction

The SELRand library contains a set of functions that return pseudo-random numbers for testing and other non-cryptographic purposes. A pseudo-random number generator is a math function that takes advantage of its previous result and large number manipulations to perform operations that are not immediately predictable. The beginning value for each use of the sequence is called the seed. Generally, the seed is created from some measurement expected to have a wide range of variance. Commonly, this can be noise on a wire, small units of time, quantity and value of user input, or some other equally unpredictable quantity. Pseudo-random number generators do tend to be cyclic in nature, but they can have very large cycle times and excellent distribution.

These functions are not intended to be used for implementing any security feature.

Supported Firmware Versions

You can use this library on any device configured using ACSELERATOR RTAC® SEL-5033 Software with firmware version R143 or higher.

Versions 3.5.0.0 and older can be used on RTAC firmware version R132 and higher.

Functions

fun_Rand (Function)

This function randomly seeds itself and returns a pseudo-random number in the range of 0 inclusive to 2^{32} exclusive.

Return Value

IEC 61131 Type	Description
UDINT	A random number between 0 inclusive to 2^{32} exclusive.

Processing

At the first call of this function it randomly seeds itself. It will continually return pseudo-random values with each use.

fun_RandRepeatable (Function)

This function always seeds itself with the same value and returns a pseudo-random number in the range of 0 inclusive to 2^{32} exclusive.

Return Value

IEC 61131 Type	Description
UDINT	A random number between 0 inclusive to 2^{32} exclusive.

Processing

At the first call of this function it seeds itself, always with the same value. It then returns the same sequence of pseudo-random values with each use.

fun_RandRanged (Function)

This function randomly seeds itself and returns a pseudo-random number between *minimum* to *maximum*, inclusive.

Inputs

Name	IEC 61131 Type	Description
minimum	UDINT	The lower limit of the return value.
maximum	UDINT	The upper limit of the return value.

Return Value

IEC 61131 Type	Description
UDINT	A random number between <i>minimum</i> and <i>maximum</i> , inclusive.

Processing

At the first call of this function, randomly seeds itself. It then continually returns pseudo-random values within the requested range.

fun_RandRangedRepeatable (Function)

This function always seeds itself with the same value and subsequently returns a pseudo-random number between *minimum* to *maximum*, inclusive.

Inputs

Name	IEC 61131 Type	Description
minimum	UDINT	The lower limit of the return value.
maximum	UDINT	The upper limit of the return value.

Return Value

IEC 61131 Type	Description
UDINT	A random number between <i>minimum</i> and <i>maximum</i> , inclusive.

Processing

At the first call of function it seeds itself, always with the same value. It then continually returns the same sequence of pseudo-random values within the requested range.

Benchmarks

Benchmark Platforms

The benchmarking tests recorded for this library are performed on the following platforms.

- SEL-3530
 - R134-V0 firmware
- SEL-3354
 - Intel Pentium 1.4 GHz
 - 1 GB DDR ECC SDRAM
 - SEL-3532 RTAC Conversion Kit
 - R132-V0 firmware
- SEL-3555
 - Dual-core Intel i7-3555LE processor
 - 4 GB ECC RAM
 - R134-V0 firmware

Benchmark Test Descriptions

All benchmark tests are based on an average of 100 calls to the function.

fun_Rand()

The posted time is the average execution time of 100 consecutive calls.

fun_RandRanged()

The posted time is the average execution time of 100 consecutive calls with a range of 1 to 100.

fun_RandRepeatable()

The posted time is the average execution time of 100 consecutive calls.

fun_RandRangedRepeatable()

The posted time is the average execution time of 100 consecutive calls with a range of 1 to 100.

Benchmark Results

Operation Tested	Platform (time in μs)		
	SEL-3530	SEL-3354	SEL-3555
fun_Rand()	1	1	1
fun_RandRanged()	2	1	1
fun_RandRepeatable()	1	1	1
fun_RandRangedRepeatable()	1	1	1

Examples

These examples demonstrate the capabilities of this library. Do not mistake them as suggestions or recommendations from SEL.

Implement the best practices of your organization when using these libraries. As the user of this library, you are responsible for ensuring correct implementation and verifying that the project using these libraries performs as expected.

Getting a Random Integer Between 10 and 50

Objective

A user knows the input for a function will vary between 10 and 50 and would like demonstrate the behavior of a function in a non-linear progression.

Assumptions

This example assumes that there is a user specified IEC 61131 function that is defined as shown in *Code Snippet 1*.

Code Snippet 1 fun_SomeFunction

```
FUNCTION fun_SomeFunction :REAL
VAR_INPUT
    num : UDINT;
END_VAR
```

Solution

The user can create the program in *Code Snippet 2* to run the function 1000 times with assorted input.

Code Snippet 2 prg_RandomWalk

```
PROGRAM prg_RandomWalk
VAR
    i : UDINT;
    results : ARRAY [1 .. 1000] OF REAL;
    randomNum : UDINT;
END_VAR

FOR i := 1 to 1000 DO
    //This gives a random number from 10 to 50;
    randomNum := fun_RandRanged(10, 51);
    //Call the function with the random number
    results[i] := fun_SomeFunction(randomNum);
END_FOR
```

Testing With Assorted Input

Objective

A user wants to vary inputs for testing but would like to be able to recreate the inputs in case something goes wrong.

Assumptions

This example assumes that there is a user-specified IEC 61131 function that is defined as shown in *Code Snippet 3*

Code Snippet 3 fun_TestFunction

```
FUNCTION fun_TestFunction :BOOL
VAR_INPUT
    num : UDINT;
END_VAR
```

Solution

The user can create the program in *Code Snippet 4* to run the function 1000 times with assorted input but a set order.

Code Snippet 4 prg_RandomTest

```
PROGRAM prg_RandomTest
VAR
  i : UDINT;
  results : ARRAY [1 .. 1000] OF REAL;
  randomNum : UDINT;
END_VAR

FOR i := 1 TO 1000 DO
  //This gives a random number;
  randomNum := fun_RandRepeatable();
  //Call the function with the random number
  results[i] := fun_TestFunction(randomNum);
END_FOR
```


Release Notes

Version	Summary of Revisions	Date Code
3.5.1.0	<ul style="list-style-type: none">▶ Allows new versions of ACSELERATOR RTAC to compile projects for previous firmware versions without SEL IEC types “Cannot convert” messages.▶ Must be used with R143 firmware or later.	20180921
3.5.0.0	<ul style="list-style-type: none">▶ Initial release.	20140811